

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property
Organization
International Bureau



(43) International Publication Date
11 November 2004 (11.11.2004)

PCT

(10) International Publication Number
WO 2004/097604 A2

(51) International Patent Classification⁷: **G06F 1/00**

[GB/GB]; Star Internet, Brighthouse Court, Barmwood, Gloucester GL4 3RT (GB).

(21) International Application Number:
PCT/GB2004/000997

(74) Agents: **AYERS, Martyn, L., S. et al.**; J.A. Kemp & Co., 14 South Square, Gray's Inn, London WC1R 5JJ (GB).

(22) International Filing Date: 8 March 2004 (08.03.2004)

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
0309464.6 25 April 2003 (25.04.2003) GB

(71) Applicant (for all designated States except US): **MES-SAGELABS LIMITED** [GB/GB]; 1270 Lansdowne Court, Gloucester Business Park, Gloucester GL3 4AB (GB).

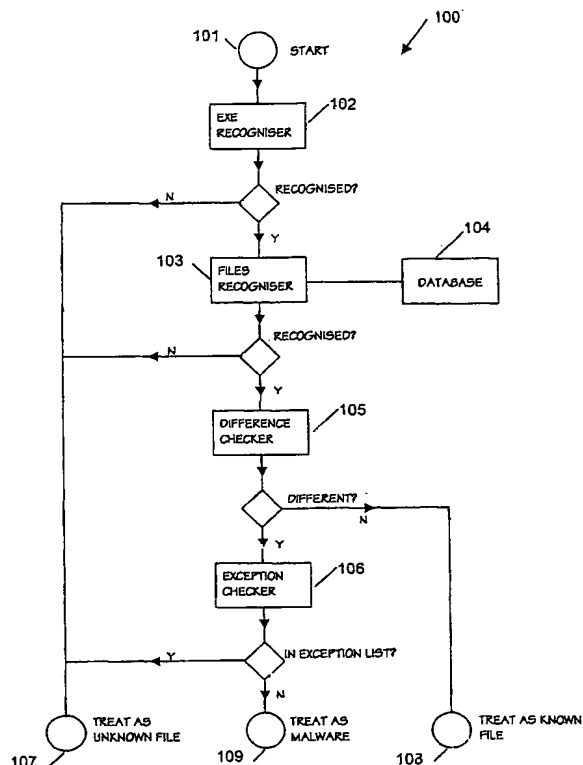
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR,

(72) Inventor; and

(75) Inventor/Applicant (for US only): **SHIPP, Alexander**

[Continued on next page]

(54) Title: A METHOD OF, AND SYSTEM FOR, HEURISTICALLY DETECTIVE VIRUSES IN EXECUTABLE CODE



(57) Abstract: In an anti-virus scanning system for computer files being transferred between computers, the number of files requiring detailed scanning is first reduced by identifying files which are instances of programs which are known and deemed to be safe. This is done by reference to a database of known executables which records characteristics which can be used as the basis for identifying a file as an unchanged instance of a known executable. Secondly, these characteristics can then also be used to identify files which are changed instances of known executables. These are extremely suspicious, since the most likely cause of change is infection by a file infecting virus, so these files are classed as likely to be malware.

WO 2004/097604 A2



GB, GR, HU, IE, IT, LU, MC, NL, PL, PT, RO, SE, SI, SK,
TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW,
ML, MR, NE, SN, TD, TG).

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Published:

— *without international search report and to be republished
upon receipt of that report*

A METHOD OF, AND SYSTEM FOR, HEURISTICALLY DETECTING VIRUSES IN EXECUTABLE CODE

The present invention relates to a method of, and system for, heuristically detecting viruses in executable code by detecting that an executable file is likely to be a previous known executable file, and that the file has been changed. This technique is especially applicable to situations where files enter a system, are checked, then leave, such as email gateways or web proxies. However, it is not intended to be limited to those situations.

The expression "virus" as used in this specification and claims is to be understood in a broad, inclusive sense encompassing any form of malware in executable code.

Increasing use of the Internet, personal computers and local- and wide-area networks has made the problem of computer viruses ever more acute.

Some internet service providers (ISP) offer anti-virus scanning, of attachments to e-mails, and file-downloads and transfers, as a value-added service to their clients. A conventional method of anti-virus scanning is to scan the file looking for patterns or sequences of bytes which have been established as being a characteristic "signature" of a known virus. However, signature-based scanning is not ideal in the rapidly changing internet environment particularly if it is used as the sole virus detection method. When an outbreak of a previously-unknown virus occurs, anti-virus specialists have first to identify a suitable signature to characterise the virus and this then has to be disseminated to anti-virus scanners in the field, all of which takes time. Another disadvantage is that file-scanning can very resource-intensive, particularly where file traffic volumes are high. The system needs to have enough processing power and file-buffering capacity to keep delays to a minimum and to cope with peak demands.

According to the present invention, there is provided an anti-virus file scanning system for computer files comprising:

a computer database containing records of known executable programs which are deemed to be uninfected and criteria by which a file being processed can be determined to be an instance of one of those programs;

means for processing a file being transferred between computers to determine whether the file matches the criteria characterising a file as an unchanged instance of a program in the database; and

means for signalling the file as known or not depending on the determination made by the processing means:

The invention also provides a method of anti-virus scanning system computer files comprising:

5 maintaining a computer database containing records of known executable programs which are deemed to be uninfected and criteria by which a file being processed can be determined to be an instance of one of those programs;

processing a file being transferred between computers to determine whether the file matches the criteria characterising a file as an unchanged instance of a program in
10 the database; and

signalling the file as known or not depending on the determination made by the processing means.

The invention is based upon the fact that a significant proportion of network traffic of executable files is made up of uninfected copies of common applications and
15 utilities such as WinZip and the like. If these can be reliably identified as such, the system need not scan them further, so reducing the processing and storage load on the system.

The invention will be further described by way of non-limitative example with reference to the accompanying drawings, in which

Figure 1 is a block diagram of the present invention.

20 Figure 1 illustrates one form of a system according to the present invention, which might be used, for example by an ISP as part of a larger anti-virus scanning system which employs additional scanning methods on files which are not filtered out as "safe" by the system of Figure 1.

The source of the files inputted to the system of Figure 1 is not material to
25 the invention. The files could, for example, be copies of attachments of e-mails being processed by an ISP en route to delivery to its customers, which have been temporarily saved to disk for anti-virus processing before the e-mail is delivered. In the case of processing file downloads or transfers, the system could be associated with a trusted proxy server which retrieves the file from an untrusted remote site, e.g. a website or FTP site,
30 saves it temporarily to disk for scanning and then delivers it to the user via whatever internet protocol he or she is using.

The nature of a file-infecting virus is such that it changes the contents of an executable file by inserting the virus code. On a file system, it is possible to detect this change by creating a checksum or hash string of the file (such as those generated by the

well-known MD5 or SHA5 checksum algorithms), and comparing it to the checksum that the file should have. If they are the same, then to an extremely high probability, the file is unchanged. If they are different, then the file has been changed.

However, this check is not possible if it is being carried out at a gateway, such as an email gateway. When the file arrives, the checksum of the file in its good state is not known – only the current checksum can be calculated. Therefore, there is no immediate method of determining that the file has been changed.

One way to get round this is to find a way of recognising the file. Once this is done, the checksum of the file in its good state can be looked up in a database. This known good checksum can then be compared with the actual checksum of the file. If they are different, the file has been changed, and special action may then be taken – this could for instance include quarantining the file, or marking it for further automatic or manual analysis. This is the method implemented by the system 100 of Figure 1, which operates according to the following algorithm.

1. A file arrives for scanning, perhaps as an email attachment, or a web download at an input 101.
2. The file is passed to an 'EXE recogniser' 102 to check if it is an executable file. If not, it is not analysed further, and is treated as an unknown file.
3. The file is then passed to a 'File recogniser' 103 to check if it is a previously known file by reference to a database 104 of known files. If not, it is not analysed further, and is signalled as an unknown file at an output 107.
4. The file is then passed to a 'Difference checker' 105 to check if it is an unchanged copy of the known file. If so, it is not analysed further, and is signalled as a known file at an output 108.
5. The file is passed to an 'Exception checker' 106 to check for known exceptions. If an exception list match is found, no further action is taken, and the file is signalled as an unknown file at the output 107.
6. The file appears to be a changed copy of a previously known executable file. It is therefore signalled as an example of file infecting malware at an output 109.

The system 101 may form part of a larger malware detection system which submits files initially to the system 101 and then, depending on the results signalled at outputs 107, 108 and 109 may submit the file to other file-scanning sub-systems if the file is signalled by output 107 to be an unknown file, treat the file as known and safe if

signalled as such by the output 108 or handle the file as malware if signalled as such at output 109. In the latter case, the file may be subject to any of the usual malware-handling actions (or any appropriate combination of them), e.g. it may be quarantined, the intended recipient may be notified that malware has been detected, it may be submitted for attention by a human operator, it may be deleted, etc.

In the case of a file signalled as "unknown" at output 107, once this has been processed further to determine whether it is in fact malware, if the determination is that it is not, the database 104 may be automatically updated with an entry for it, so that the system 101 can treat subsequent instances of it as a known file.

One way of implementing the EXE recogniser 102 is simply by comparing the first few bytes of the file with the published specifications for the particular format.

For instance, here is a simplistic example of an algorithm for determining if a file is likely to be a Windows PE file.

Read in first 2 bytes. If these are not 'MZ' then stop

Read in another 58 bytes.

Read in 4 bytes into variable x (treating using intel byte-ordering)

Seek to offset x in file

Read in 4 bytes

If bytes are P E \0 \0, then file is likely to be a Windows PE file

Recognisers can be created for each additional executable file format as desired.

The file recogniser 1 can use various strategies for determining if the file is a known file. For instance, pattern matching scanning techniques can be used. This is similar to the techniques used by virus scanners to identify known malware – except in this case to identify known good files instead. Another technique is to split the file into areas suggested by analysing the structure of the file, and checksumming each of the areas. These can then be compared with checksums in the database 104 which have been generated by doing the same analysis on known files. Unless the virus modifies every single area, there will be at least one match which will then identify the file.

A simplistic implementation of the difference checker 105 could checksum the entire file, and compare this with the checksum the known file should have.

It is possible that occasional false positives could arise. For instance, although unlikely, two different files could have the same checksum. Another scenario is a packer compressing part of a file, but leaving others uncompressed. The exception checker

106 can be designed to accommodate these situations. For instance, in the case of the packed files, all files packed by the particular packer exhibiting the behaviour can be ignored. It will be apparent from Figure 1 that this treatment of packers does not constitute a "hole" in the system from the point of view of letting viruses through. The file is only
5 processed by the exception checker if it has already been determined that the file does not correspond to an unchanged version of a known file, and what the exception checker does is to signal whether the file a) should be considered as malware, if it is not in the exception list or b) is unknown and should therefore be subject to further virus-detection processing.

As well as using the system 101 as a stand-alone virus detection algorithm,
10 it can be combined with systems implementing other techniques as part of a larger system. For instance, programs flagged as malware by the system 101 may be allocated a certain score, or variety of scores depending which tests pass and fail. Scores may also be assigned using other heuristic techniques, and only if the total score passes some limit is the program flagged as viral.

15 The system 101 can also be used as an indicator for program files which may need further analysis. For instance, programs flagged as known files at an output 108 may not need further analysis, since they have already been flagged as 'safe'. Programs flagged at output 107 as unknown files may need further analysis.

20 The difference checker 105 can also be adapted to cope with files that are expected to change. For instance, a self extracting ZIP file may have a potentially different ZIP bound to it every time. Some programs also carry within themselves registration keys, and these may be different for every user. In such cases it is possible to create a checksum of the parts of the file that are invariant each time, and use that. For instance, the checksum may be created from bytes 0x0000 to 0x0A00, and from 0x0A73 to 0x3000.

25 If the difference checker 105 is modified in this way, then the files it matches will not necessarily be treated as known files needing no further analysis. For instance, if a self extracting ZIP file is recognised, then the files inside the ZIP archive will need extracting and analysing before the file can be considered safe. Each known file will therefore need an associated status which flags whether further processing is or is not
30 required.

CLAIMS

1. An anti-virus file scanning system for computer files comprising:
 - a) a computer database containing records of known executable programs which are deemed to be not malware and criteria by which a file being processed
5 can be determined to be an instance of one of those programs;
 - b) means for processing a file being transferred between computers to determine whether the file matches the criteria characterising a file as an unchanged instance of a program in the database; and
 - c) means for signalling the file as known or not depending on the
10 determination made by the processing means.
2. A system according to claim 1 and including:
 - d) means for processing an inputted file to determine whether it is considered to be, or considered possibly to be, infected with a virus, and wherein, in operation of the system, a file is subjected to processing by the means d) unless the file is
15 signalled as safe by the signalling means c).
3. A system according to claim 1 or 2 wherein the processing means b) comprises
 - b1) a file recogniser for determining whether the file being processed is an instance of a known file and
 - 20 b2) a difference checker for checking whether the file is an unchanged version of that known file.
4. A system according to claim 3 wherein the file recogniser includes means for checking the contents of the file being processed for the presence of at least one characteristic signature associated with a file which is considered to be known and
25 uninfected.
5. A system according to any one of the preceding claims wherein the processing means b) includes means for generating a checksum for the entire file under

consideration or for at least one selected region thereof, and means for comparing the checksum or checksums with those of entries in the database.

6. A system according to any one of the preceding claims and including an exception list handler for determining, in relation to a file which the processing means b) has determined is not a known file, whether that file has characteristics matching an entry in an exception list of files, and the signalling means is operative to signal the file as malware if it is not in the exception list or as unknown if it is.

7. A method of anti-virus scanning system computer files comprising:

maintaining a computer database containing records of known executable programs which are deemed to be uninfected and criteria by which a file being processed can be determined to be an instance of one of those programs;

processing a file being transferred between computers to determine whether the file matches the criteria characterising a file as an unchanged instance of a program in the database; and

signalling the file as known or not depending on the determination made by the processing means.

8. A method according to claim 7 and including:

processing an inputted file to determine whether it is considered to be, or considered possibly to be, infected with a virus, and wherein, in operation of the system, a file is subjected to processing by the step d) unless the file is signalled as safe by the signalling step c).

9. A method according to claim 7 or 8 wherein the processing step b) uses b1) a file recogniser for determining whether the file being processed is an instance of a known file and b2) a difference checker for checking whether the file is an unchanged version of that known file.

10. A method according to claim 9 wherein the file recogniser includes means for checking the contents of the file being processed for the presence of at least one

characteristic signature associated with a file which is considered to be known and uninfected.

11. A method according to any one of claims 7 to 10 wherein the processing step b) includes generating a checksum for the entire file under consideration or for at least one selected region thereof, and comparing the checksum or checksums with those of entries in the database.

12. A method according to any one claims 7 to 11 and including using an exception list to determine, in relation to a file which the processing step b) has determined is not a known file, whether that file has characteristics matching an entry in an exception list of files, and wherein, in the signalling steps, the file is signalled as malware if it is not in the exception list or as unknown if it is.

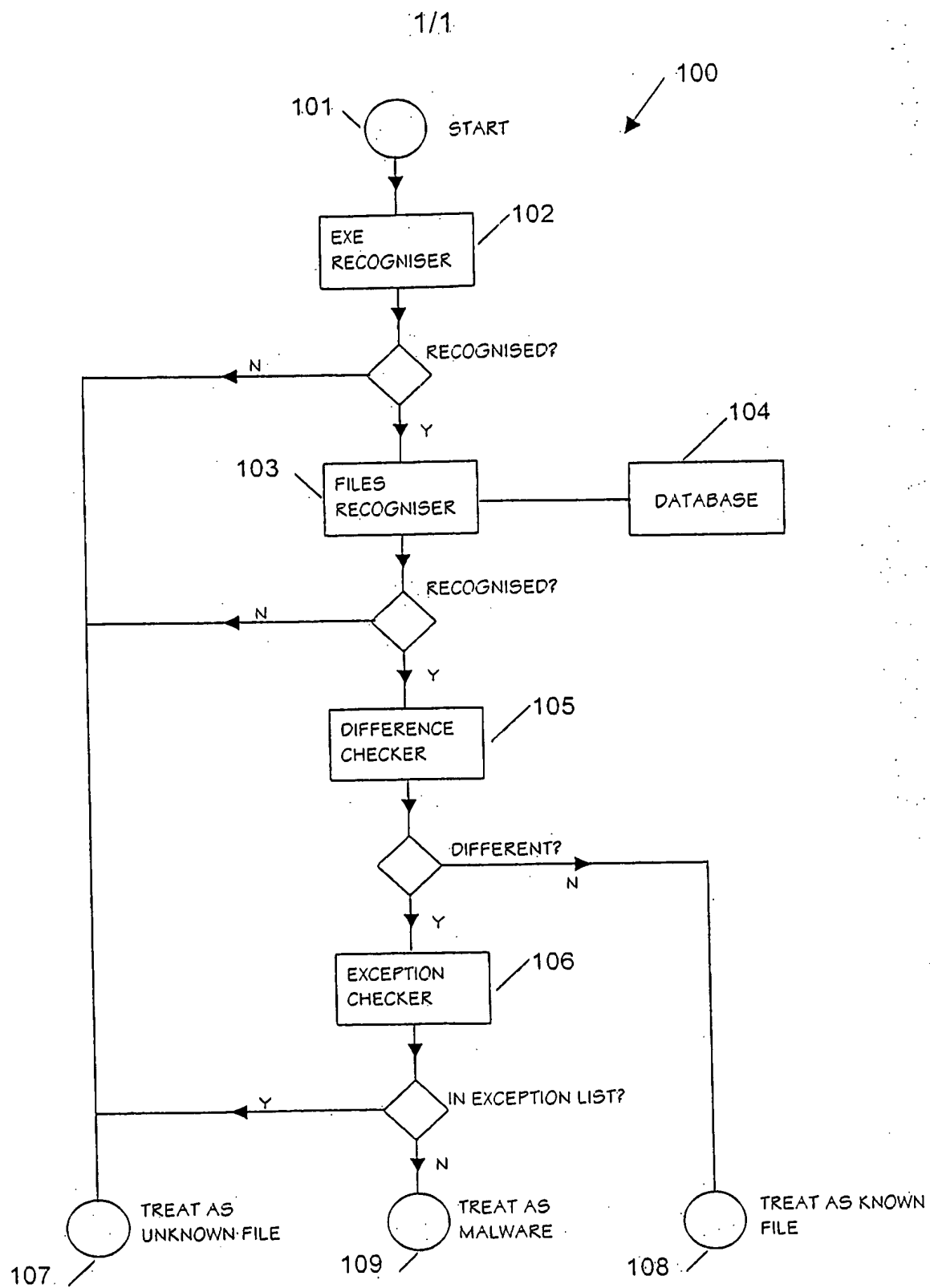


Fig.1

(19) World Intellectual Property
Organization
International Bureau



(43) International Publication Date
11 November 2004 (11.11.2004)

PCT

(10) International Publication Number
WO 2004/097604 A3

(51) International Patent Classification⁷: **G06F 1/00**

[GB/GB]; Star Internet, Brighthouse Court, Barmwood, Gloucester GL4 3RT (GB).

(21) International Application Number:
PCT/GB2004/000997

(74) Agents: **AYERS, Martyn, L., S. et al.**; J.A. Kemp & Co., 14 South Square, Gray's Inn, London WC1R 5JJ (GB).

(22) International Filing Date: 8 March 2004 (08.03.2004)

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
0309464.6 25 April 2003 (25.04.2003) GB

(71) Applicant (for all designated States except US): **MES-SAGELABS LIMITED** [GB/GB]; 1270 Lansdowne Court, Gloucester Business Park, Gloucester GL3 4AB (GB).

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR,

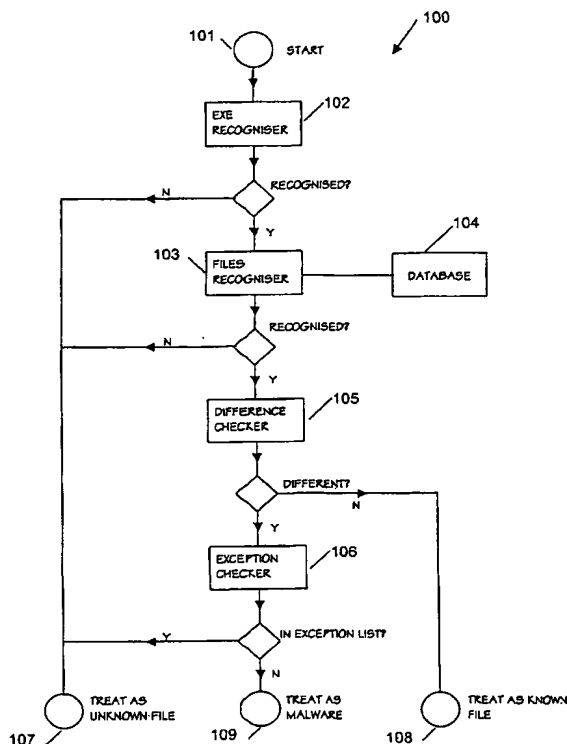
(72) Inventor; and

(75) Inventor/Applicant (for US only): **SHIPP, Alexander**

[Continued on next page]

(54) Title: A METHOD OF, AND SYSTEM FOR, HEURISTICALLY DETECTIVE VIRUSES IN EXECUTABLE CODE

(57) Abstract: In an anti-virus scanning system for computer files being transferred between computers, the number of files requiring detailed scanning is first reduced by identifying files which are instances of programs which are known and deemed to be safe. This is done by reference to a database of known executables which records characteristics which can be used as the basis for identifying a file as an unchanged instance of a known executable. Secondly, these characteristics can then also be used to identify files which are changed instances of known executables. These are extremely suspicious, since the most likely cause of change is infection by a file infecting virus, so these files are classed as likely to be malware.





GB, GR, HU, IE, IT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

(88) Date of publication of the international search report:
10 March 2005

Published:

- with international search report
- before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

INTERNATIONAL SEARCH REPORT

PCT/GB2004/000997

A. CLASSIFICATION OF SUBJECT MATTER

IPC 7 G06F1/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	GB 2 378 015 A (NETWORKS ASSOC TECH INC) 29 January 2003 (2003-01-29) page 1, line 12 - page 4, line 2 page 5, line 5 - page 10, line 15 -----	1-12
X	EP 0 813 132 A (IBM) 17 December 1997 (1997-12-17) page 1, line 1 - page 1, line 44 page 2, line 58 - page 5, line 24 -----	1,7
X	WO 02/33525 A (CHUANG SHYNE SONG) 25 April 2002 (2002-04-25) page 4, line 26 - page 5, line 21 page 6, line 26 - page 9, line 26 -----	1,2,7,8
A	EP 1 291 749 A (NETWORKS ASSOC TECH INC) 12 March 2003 (2003-03-12) the whole document -----	1-12



Further documents are listed in the continuation of box C.



Patent family members are listed in annex.

* Special categories of cited documents:

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier document but published on or after the international filing date
- *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- *O* document referring to an oral disclosure, use, exhibition or other means
- *P* document published prior to the international filing date but later than the priority date claimed

- *T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- *&* document member of the same patent family

Date of the actual completion of the international search

21 December 2004

Date of mailing of the international search report

04/01/2005

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Pinto, R.M.

INTERNATIONAL SEARCH REPORT

PCT/GB2004/000997

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
GB 2378015	A	29-01-2003	US	2003023865 A1	30-01-2003
EP 0813132	A	17-12-1997	US	5825877 A	20-10-1998
			EP	0813132 A2	17-12-1997
			JP	10083310 A	31-03-1998
			KR	267872 B1	16-10-2000
WO 0233525	A	25-04-2002	AU	9620501 A	29-04-2002
			WO	0233525 A2	25-04-2002
			US	2004039921 A1	26-02-2004
EP 1291749	A	12-03-2003	US	2003046558 A1	06-03-2003
			EP	1291749 A2	12-03-2003